

**CP/M plus**

**auf dem**

**PROF-80**

 **CONITEC**  
J. H. Christian Lotter KG · Datensysteme

Grafenstraße 31  
Postfach 1 106 22  
D-6100 Darmstadt

## CP/M plus für den PROF-80

Zunächst eine Vorbemerkung: CP/M plus gibt es in zwei Versionen, nämlich einer unbankten Version, die maximal 64K RAM verwalten kann, und einer gebankten Version, die mehr als 64K RAM ausnutzen kann. Diese Blätter beziehen sich ausschließlich auf die gebankte Version von CP/M plus.

### Was leistet CP/M plus ?

CP/M plus ist wie seine Vorgänger (verbreitete Versionen sind CP/M 1.4 und CP/M 2.2) ein Einbenutzer Betriebssystem für Mikroprozessoren mit 8080-Befehlssatz ( z.B.: Intel 8080, Intel 8085, Zilog Z80).

CP/M plus ist eine standardisierte Schnittstelle zwischen Anwenderprogrammen (z.B. Textverarbeitung) und der jeweiligen Hardware (z.B. PROF-80). Ein Programm, das auf einem beliebigen Computer mit CP/M plus Betriebssystem erstellt wurde, ist, sofern Ein-Ausgaben nur über das Betriebssystem geführt werden, ohne Änderung auf allen anderen CP/M plus Computern lauffähig.

CP/M plus regelt die Ein-Ausgabe von Zeichen, verwaltet die Daten auf der Floppy, verwaltet den zur Verfügung stehenden Arbeitsspeicher und einiges mehr.

Dies sind alles Eigenschaften, die auch schon die älteren CP/M Versionen aufweisen konnten. Es drängt sich deshalb die Frage auf, was an CP/M plus neu ist. Nun, gegenüber seinen Vorgängern, bietet CP/M plus unter anderem folgende Neuerungen:

- Unterstützung einer Echtzeituhr (z.B. kann bei Dateien das Datum und die Uhrzeit des letzten Zugriffs oder der letzten Änderung automatisch gespeichert werden)
- Es werden mehr als 64K Arbeitsspeicher genutzt; der Speicherbereich für Anwenderprogramme hat sich

allerdings nur unwesentlich erhöht (üblicherweise 60K)

- durch eine große Zahl von Sektor-Puffern verringern sich die Diskettenzugriffe
- die Datensicherheit hat sich erhöht
- Dateien können gegen unberechtigten Zugriff geschützt werden (Passwords)
- Erweiterung der Betriebssystem-Funktionen durch RSX-Module (RSX: Resident System Extension)
- der Bedienkomfort hat sich wesentlich erhöht (z.B. editieren der Eingabezeile, Wiederholung der zuletzt getippten Zeile, Mehrfach-Anweisungen)
- der mitgelieferte "Help-Befehl" erspart so manches Blättern im Handbuch

Darüberhinaus war Digital Research nicht knausrig, was die mitgelieferten Hilfsprogramme betrifft; z.B. gehören der Symbolische Debugger SID oder die Assembler MAC und RMAC zur Standard Lieferung.

### Die Struktur von CP/M plus

Wie bei seinem Vorgänger besteht der Kern von CP/M plus aus einem hardwareunabhängigen Teil (BDOS: Basic Disk Operating System) und einer Sammlung von hardwareabhängigen Routinen (BIOS: Basic Input Output System), die an den jeweiligen Rechner angepaßt sein müssen.

BDOS und BIOS sind in zwei Einheiten aufgeteilt: Die eine (kleinere) Einheit befindet sich immer im Adressbereich des Mikroprozessors, gleich welche Speicherbank eingeschaltet ist (Common-Bereich). Der zweite Teil von BDOS und BIOS liegt im

gebankten Teil des Arbeitsspeichers und wird nur bei Bedarf eingeschaltet.

Ein CCP (Console Command Prozessor) als Schnittstelle zum Benutzer ist ebenfalls vorhanden. Im Gegensatz zu dem CCP des CP/M 2.2 bildet der CP/M plus CCP keine Einheit mit dem BDOS-BIOS-Kern sondern wird nach jedem Warmboot als gewöhnliches TPA-Programm (TPA: Transient Programm Area) in den Arbeitsspeicher geladen. Beim Laden eines Anwenderprogramms wird der CCP automatisch überschrieben.

### Die Kompatibilität zwischen CP/M plus und CP/M 2.2

Sofern nur die Standard BDOS-Schnittstelle verwendet wird, ist CP/M plus voll aufwärtskompatibel zu CP/M 2.2. Probleme gibt es lediglich mit einer kleinen Anzahl von Programmen, die die direkten BIOS-Einsprünge benutzen und womöglich sogar noch den BDOS-Teil überschreiben.

### Die Speicheraufteilung

Ein wesentlicher Unterschied zwischen CP/M plus und CP/M 2.2 besteht darin, daß CP/M plus in der Lage ist, mehr Speicher zu nutzen, als die Z80-CPU überhaupt adressieren kann. Bild 1 zeigt die grundsätzliche Hardwarestruktur, die CP/M plus in der gebankten Version benötigt. Der gesamte Arbeitsspeicher wird in einzelne Banks und einen sogenannten Commonbereich aufgeteilt. Der Commonbereich befindet sich immer im logischen Adressbereich der CPU, wohingegen die einzelnen Banks umgeschaltet werden können.

CP/M plus verwaltet die einzelnen Speicherteile folgendermaßen:

Bank 1 und Commonbereich bilden zusammen den TPA-Bereich für Anwenderprogramme, wie er schon vom CP/M 2.2 bekannt ist. Damit der TPA-Bereich lückenlos ist, muß sich an die obere Grenze von Bank 1 der Commonbereich anschließen. Weiterhin muß Bank 1 bei Adresse 0 beginnen. Oberhalb des TPA-Bereiches liegt der ungebankte Teil des Betriebssystems mit seinem bekannten BDOS-Einsprungpunkt.

Im oberen Adressbereich von Bank 0 liegt der restliche (und größere) Teil des Betriebssystems der vom ungebankten Teil bei Bedarf eingeschaltet wird.

Der Rest von Bank 0 und alle weiteren Bänke dienen als Zwischenspeicher für physikalische Diskettensektoren. Bevor ein Diskzugriff stattfindet schaut das Betriebssystem erst einmal nach, ob sich der entsprechende Sektor schon im RAM befindet und kopiert ihn gegebenenfalls an die gewünschte Stelle. Durch diese Methode verringert sich die Anzahl der wirklichen Diskettenzugriffe, und die Verarbeitungsgeschwindigkeit der einzelnen Programme steigt.

Bild 2 zeigt die spezielle Speicheraufteilung für den PROF-80. Die oberen 4K von Bank 1, die parallel zum Commonbereich liegen können von CP/M plus nicht verwendet werden. Wir haben diese 4K deshalb zum Zwischenspeichern des CCP genutzt. Bei einem Warmboot wird dieser Bereich einfach in die TPA kopiert, dadurch entfällt der sonst übliche Diskettenzugriff.

### Das CP/M plus BIOS

Bild 3 zeigt den BIOS-Einsprung-Vektor, der gegenüber seinem Vorgänger von 17 auf 33 Einsprungstellen erweitert wurde. Die unteren 17 Einsprungstellen sind mit denen des CP/M 2.2 identisch, allerdings mit einem wesentlichen Unterschied: bei allen Diskettenzugriffen unter CP/M plus werden physikalische Sektoren, die je nach Formatierung

unterschiedlich groß sein können, übertragen. Die Umwandlung in 128 Byte große logische Records (Blocking/Deblocking) wird bei CP/M plus vom BDOS übernommen.

Durch die Erweiterung auf 33 BIOS Einsprungstellen werden im wesentlichen die folgenden neuen Funktionen ermöglicht:

- Umschalten der Speicherbänke für Programmcode und Diskzugriffe (SELMEM, SETBNK)
- Kopierroutinen, die den Z80-LDIR-Befehl oder einen DMA-Baustein ausnutzen können (MOVE, XMOVE)
- Lesen und Setzen einer Echtzeit-Uhr (TIME)
- Der Status aller Zeichen- Ein- Ausgabeeinheiten kann abgefragt werden, dadurch wird die Zuordnung zwischen logischen E/A-Kanälen und physikalischen Treibern wesentlich flexibler. (CONOST, AUXIST, AUXOST)

Neben der BIOS-Sprungleiste besteht noch eine zweite Einrichtung, mit der BDOS und BIOS kommunizieren können: der sogenannte System Control Block (SCB).

Der SCB ist ein reines Datenfeld und besteht aus etwa 100 Bytes; er enthält unter anderem folgende Angaben:

- Zuordnungsvektoren zwischen logischen und physikalischen E/A-Einheiten.
- Datenfelder zum Auslesen und Setzen der Echtzeituhr
- Angaben über die Behandlung von Disk-Fehlern (z.B. ob vom BIOS eine Fehlermeldung ausgegeben werden soll)
- Momentan eingeloggtes Laufwerk

Der SCB wird bei der Systemgenerierung (mittels GENCPM) in den BDOS-Bereich eingebunden. Das BIOS kann auf alle SCB-

Daten über globale Variablennamen zugreifen.

### BIOS-Module und Systemgenerierung

Beim CP/M 2.2 befanden sich alle BIOS-Routinen in einer einzigen Assembler-Datei. Ab einer gewissen Größe ist eine solche Datei schwierig zu warten. Will man zum Beispiel in ein bestehendes CP/M 2.2 BIOS Treiber für eine Harddisk einbauen, dann ist die Wahrscheinlichkeit recht groß, daß man in vorher funktionierende Systemteile Fehler, einbaut.

Digital Research machte deshalb den Vorschlag, das BIOS in einzelne Module aufzuteilen, so daß es möglich ist, einzelne Treiber unabhängig vom restlichen BIOS zu verändern oder zu erweitern. Will man z.B., wie oben angedeutet eine Harddisk einbauen, so schreibt man den zugehörigen Treiber in eine eigene Assembler-Datei, assembliert den neuen Treiber und bindet diesen an die alten BIOS Teile.

Das CP/M-plus-BIOS des PROF-80 besteht aus den folgenden Modulen:

BIOSKRNL.ASM ist das sogenannte Root-Modul. Es enthält den BIOS-Einsprungvektor und verzweigt auf die anderen Module.

BOOT.ASM, erledigt die Systeminitialisierung (außer Disk und Zeichen-E/A Initialisierung), lädt den CCP beim Kaltstart und kopiert den CCP bei einem Warmstart.

CHARIO.ASM enthält alle Treiber und Initialisierungs-Routinen für die Zeichen Ein-Ausgabe.

MOVE.ASM ist für Speicherkopier-Routinen und Bank-Auswahl verantwortlich.

DRVTBL.ASM ordnet den logischen Laufwerken physikalische

Dieser Teil zeigt die Verfahrensweise auf mit der die 3.00  
geändert werden kann.

1. Eventuelle neue Treiber für Floppy-Laufwerke oder Winchester-Laufwerke sowie EPROM-Floppys oder CMOS-Floppys werden als separates Modul im Assembler geschrieben, und mit KMAC Assembliert.  
Es ist darauf zu achten das die notwendigen LIB-Dateien (z.B. LIBE) ebenfalls auf der Diskette vorhanden sind.
2. In der DRZFILE.mali müssen die neuen Laufwerke an der entsprechenden Stelle eingetragen werden.  
Beispiel: FD1510 soll Laufwerk 1 sein, FD1510 erscheinend dann an erster Stelle in der Parameterdatei DRZFILE.mali.
3. Sind alle malifiles assembliert werden diese mit dem Loader (LIB.LDR) zu einem neuen BIN-BIOS 3.00% zusammengeführt.  
LIB BIN BIOS 3.00% BIN BIOS KBI, SCD, DDD1, CHAP10, DDD2, DDD3, DDD4, DDD5, DDD6, DDD7, DDD8, DDD9
4. Das GENBIN wird dann die neue Betriebssystem heruntergeladen.  
Es ändert das File entsprechend anwendung.  
Eventuell müssen die Laufwerkparameter geändert werden (Parameter etc.).

VORSTICH des malifiles auf der EPROM-Floppy installiert.  
CBI 3.00 System hat folgende Laufwerkbezeichnungen:

- A - EPROM-Floppy
- B - Laufwerk 0 (1-4 Zoll 5 1/4 inch)
- C - Laufwerk 1 (8 Zoll Single Density - Single Sides)
- D - 1111
- E - 1111
- F - CMOS Floppy

Der Loader auf den Systemspuren der EPROM-Floppy lädt CBI 3.00% von der EPROM-Floppy. Der Loader hier nur steht unter CBI DR4.COM vorgefertigt auf Diskette und nur wenn von PIPPI neu geladen werden.

Treiber zu.

FDPROFBK.ASM erledigt die Disketten Ein-Ausgabe für den PROF-80.

SCB.ASM (System Control Block, siehe oben)

Die einzelnen Module werden getrennt assembliert (mit RMAC) und mittels einem Linker (LINK) zu dem BNKBIOS3.SPR-File zusammengebunden. Der RMAC-Assembler kann zwischen verschiedenen internen Programmzählern umschalten, dadurch ist es möglich, bestimmte BIOS-Teile dem Commonbereich und dem gebankten Bereich zuzuordnen.

Die eigentliche Systemgenerierung übernimmt das Hilfsprogramm GENCPM. Es erzeugt aus den Dateien RESBDOS.SPR (BDOS-Teil der im Commonbereich liegt), BNKBDOS.SPR (gebankter BDOS Teil) und BNKBIOS3.SPR das lauffähige Betriebssystem und legt es in der CPM3.SYS Datei ab.

### Kaltstart von CP/M plus

CP/M plus wird immer von Laufwerk A geladen. Dies kann auf drei Arten geschehen:

- Nach dem Einschalten durch Drücken irgendeiner Taste (außer Return).
- Vom Monitor aus mit dem Boot-Befehl.
- Wenn Jumper J4 auf Stellung 4-6 steht, wird nach dem Einstecken einer Diskette in A CP/M plus geladen.

Das Laden selbst erfolgt in mehreren Stufen, um die sich der Benutzer allerdings nicht zu kümmern braucht.

1.) Der Monitor paßt seine Disketten-Parameter an das Laufwerk A und das Format der Spur 0 an.

2.) Der Monitor liest den ersten Sektor von Spur 0, auf dem sich eine 128 Byte lange Boot-Routine befindet, in den Arbeitsspeicher nach Adresse FA00h und kopiert danach den Bereich FA00h-FA7Fh nach FD80h. Jetzt verzweigt der Monitor nach FD80h.

3.) Die Programmkontrolle liegt nun bei der Boot-Routine, deren Aufgabe es ist einen speziellen CP/M-plus-Lader zu laden, der sich ähnlich wie CP/M 2.2 auf den Systemspuren befindet.

4.) Die Programmkontrolle wird an den CP/M-plus-Lader übergeben. Der CP/M-plus-Lader lädt die (normale CP/M) Datei CPM3.SYS in den Arbeitsspeicher und gibt eine Tabelle über die Speicherbelegung des CP/M plus Systems aus. (Die Datei CPM3.SYS enthält das eigentliche CP/M plus bestehend aus BDOS und BIOS und Informationen darüber, wo das System geladen werden soll).

5.) CP/M plus befindet sich nun im Arbeitsspeicher und der Kaltstart-Einsprung des BIOS erhält die Programmkontrolle. Das BIOS gibt eine Meldung an die Konsole aus und lädt die Datei CCP.COM in den TPA-Bereich. Eine Kopie des CCP wird in einem reservierten RAM-Bereich (Bank1 F000h-FC7Fh) abgelegt.

6.) Der CCP erhält die Programmkontrolle vom BIOS und meldet sich mit dem gewohnten 'A>' auf dem Bildschirm. CP/M plus ist einsatzbereit.

Zusätzlich zur oben genannten Methode besteht noch die Möglichkeit, CP/M plus von CP/M 2.2 aus zu laden. Das Programm LADER3.COM lädt, wie in Schritt 4 angegeben, die Datei CPM3.SYS in den Arbeitsspeicher. Der Rest läuft wie

bereits beschrieben ab.

### Kopieren des CP/M plus Systems

Wie aus dem Vorangegangenen ersichtlich ist, werden vier Programme benötigt, um das CP/M plus System von der Diskette in den Arbeitsspeicher zu laden:

- eine Boot-Routine, die sich auf Spur 0 / Sektor 1 befinden muß
- einen CP/M-plus-Lader (Länge etwa 5,5K), der sich ab Spur 0 / Sektor 2 fortlaufend auf den Systemspuren befindet
- das eigentliche CP/M plus Betriebssystem, daß sich in der Datei CPM3.SYS befindet
- der Console Comand Prozessor (CCP.COM)

Die Boot-Routine und der CP/M-plus-Lader werden, wie vom CP/M 2.2 her gewohnt, mit dem Programm CSYSGEN.COM von der Mutterdiskette auf andere Disketten kopiert. Die Formate der beiden Disketten können unterschiedlich sein; CSYSGEN paßt sich automatisch daran an.

CPM3.SYS und CCP.COM werden mit dem Standard-Programm PIP kopiert.

Nachdem dies vier Programme kopiert wurden, ist die Diskette 'bootfähig'. Es ist allerdings empfehlenswert, außer diesen unbedingt notwendigen Programmen, noch einige Hilfsprogramme zu kopieren, da dadurch der Arbeitskomfort wesentlich gesteigert wird.

### Formatieren

Das Formatieren von Disketten geschieht, wie unter CP/M 2.2 gewohnt, mit dem Programm CINIT.

Folgende Parameter können frei gewählt werden:

- Mini- oder Maxi-Format
- Single- oder Double-Density
- ein- oder zweiseitig
- Sektorgröße
- Anzahl der Spuren

Bei der Auswahl ist zu beachten, daß standardmäßig nur die in Tabelle 2 aufgeführten Formate vom BIOS verarbeitet werden können.

Hinweis für CP/M 2.2 Besitzer: gegenüber der CP/M 2.2 Anpassung haben wir ein häufig verwendetes Format (Format 10, Tabelle 2) in die Standard-Anpassung aufgenommen; dieses neue Format kann erst ab CINIT Version 2.3 formatiert werden (bei älteren CINIT Versionen werden 18 statt 16 Sektoren/Spur formatiert).

#### Ein-/Ausgabe-Einheiten

CP/M plus verfügt über 5 logische Zeichen Ein-/Ausgabe-Kanäle:

- Konsoleneingabe (CONIN)
- Konsolenausgabe (CONOUT)
- zusätzlicher Eingabekanal (AUXIN) z.B. für ein Modem
- zusätzlicher Ausgabekanal (AUXOUT) z.B. für ein Modem
- Drucker-Ausgabe-Kanal (LST)

Diesen logischen Kanälen können Physikalische Treiber zugeordnet werden. Beim PROF-80 stehen ohne BIOS-Änderung folgende Treiber zur Verfügung.

- Duplex Schnittstelle auf PROF-80 "DUPLEX" (Ein-/Ausgabe)
- Simplex Schnittstelle auf PROF-80 "SIMPLX" (Ausgabe)
- GRIP Tastatur "GRIP" (Eingabe)
- GRIP Bildschirm "GRIP" (Ausgabe)
- GRIP Drucker "SPOLER" (Ausgabe)
- User1 "USER1" (Ein-/Ausgabe)
- User2 "USER2" (Ein-/Ausgabe)

Die Zuordnung der Treiber zu den logischen Kanälen erfolgt über fünf 16-Bit-Vektoren. Zu jedem logischen Kanal gehört ein Vektor; die einzelnen Bits dieses Vektors representieren die zugeordneten physikalischen Treiber. Mit einem High-Bit wird angezeigt, daß der entsprechende Treiber über den jeweiligen logischen Kanal angesprochen werden kann.

Es ist möglich einem logischen Kanal mehrere physikalische Treiber zuzuordnen (man kann z.B. mehrere Terminals gleichzeitig dem Konsolenkanal zuordnen).

Tabelle 1 gibt einen Überblick über die Zuordnung der E/A-Kanäle nach dem Booten.

Während des Betriebs können die Zuordnungen über das DEVICE-Programm beliebig geändert werden.

Die Baudraten der Duplex- und der Simplex-Schnittstelle

können ebenfalls über das DEVICE-Programm eingestellt werden. Es stehen insgesamt 15 Baudrate zwischen 50 und 19200 Baud zur Verfügung, wobei die 19200 Baud allerdings nur näherungsweise erreicht werden (maximal 7% Abweichung).

Einstellung nach dem Booten:

Simplex	1200 Baud				
Duplex	Baudrate	wie	mit	J5	eingestellt

### Diskettenformate

Das PROF-80 CP/M-plus-BIOS kann standardmäßig 10 verschiedene Diskettenformate bearbeiten; Tabelle 2 gibt darüber Auskunft.

Das BIOS erkennt die verschiedenen Formate und paßt sich automatisch daran an. Zu den bestehenden 10 Formaten kann der Anwender beliebig viele eigene Formate einhängen (oder bestehende abändern).

### Einbau eigener Diskettenformate

Tabelle 2 zeigt die Diskettenformate, die ohne BIOS-Änderung vom PROF-80 bearbeitet werden können. Alle diese Formate können gleichzeitig auf beliebigen Laufwerken benutzt werden, soweit das Laufwerk dies physikalisch zuläßt (8" Disketten passen nun mal nicht in 5" Laufwerke). Nach jedem Disk-Reset oder nach einem fehlgeschlagenen Diskzugriff paßt sich PROF-80 automatisch an die Disketten, die sich gerade in den Laufwerken befinden, an.

Für die Meisten dürften wohl die in Tabelle 1 aufgeführten Formate ausreichen und eine Änderung des BIOS somit überflüssig sein. Will man jedoch mit Computern, die keines dieser

Formate bearbeiten können, Daten austauschen oder "exotische" Laufwerke anschließen (z.B. die neuen 1,6 M Byte Minilaufwerke), dann ist eine Änderung des BIOS unumgänglich.

Die Anpassung an eigene Diskettenformate ist relativ einfach, da nur neue Tabellen in das BIOS eingetragen werden müssen. Zum besseren Verständnis darüber, wie sich das BIOS automatisch an ein Diskettenformat anpaßt, einige Einzelheiten:

Über den BIOS-Einsprung SELDSK wählt das BDOS ein neues Laufwerk aus und teilt gleichzeitig mit, ob dies das erste Mal nach einem Disk-Reset der Fall ist. Wenn dieses Laufwerk zum ersten Mal ausgewählt wird, geht das BIOS davon aus, daß sich das Format der Diskette ändern konnte und testet das Diskettenformat neu. Jetzt wird in der DISK\$TYPE\$LIST (Tabelle 3) nachgeschaut, ob dieses Format verarbeitet werden kann. Kann es verarbeitet werden (nämlich dann, wenn das Byte "Disk Typ" und die Anzahl der physikalischen Sektoren/Spur mit dem erkannten Format übereinstimmen), dann wird die Typ-Nummer dieses Formates (ergibt sich aus dem relativen Eintrag in der DISK\$TYPE\$LIST) in einer Merkwelle abgelegt und die Adressen des zugehörigen Disk Parameter Blocks (DPB) und der Sektor-Skew-Tabelle in den Disk Parameter Header (DPH) des angewählten Laufwerks eingetragen. Danach arbeitet SELDSK so weiter, als wäre das Laufwerk schon mehrmals ausgewählt worden.

Kann das Diskettenformat nicht erkannt werden oder ist es nicht in der DISK\$TYPE\$LIST eingetragen, wird dies dem BDOS durch 0000h in Registerpaar HL, gemeldet und die Fehlermeldung "CP/M error on x: select" erscheint.

Wurde das Laufwerk nach dem letzten Disk-Reset schon mindestens einmal ausgewählt, dann wird davon ausgegangen, daß sich das Format nicht geändert hat.

Wie wird nun ein neues Format in das BIOS eingetragen?

Zuerst muß genau überlegt werden, wie das neue Format aussehen soll, dann entwickelt man Byte 0-1 der DISK\$TYPE\$LIST. Diese beiden Bytes dienen der Unterscheidung zu anderen Formaten und dürfen sich mit keinen bestehenden Einträgen in der DISK\$TYPE\$LIST decken. Falls dies doch der Fall sein sollte, kann nur der niedrigste Eintrag verarbeitet werden.

Zu Bit 7 von Byte 0 der DISK\$TYPE\$LIST ist noch etwas zu sagen: Bei 8" Laufwerken wird zwischen Single-Sided- und Double-Sided-Disketten durch die Lage des Indexlochs unterschieden. Je nach dem ob es sich um eine einseitige Diskette oder um eine Zweiseitige handelt, ist dieses Signal aktiv oder nicht. Bei 8" Disketten verwendet man dieses Format sinnvollerweise zur Unterscheidung zwischen ein- und zweiseitigen Disketten. 5" Laufwerke liefern das Two-Sided-Signal nicht. Die Unterscheidung zwischen ein- und zweiseitigen Disketten geschieht bei unserer Diskanpassung dadurch, daß alle einseitigen 5" Disketten eine Sektorgröße von 1024 und alle zweiseitigen 5" Disketten eine Sektorgröße von 512 haben müssen (ob die Diskette ein- oder zweiseitig beschrieben wird hängt einzig und allein von dem zugehörigen DPB ab). Da bei 5" Laufwerken das Two-Sided-Signal unbenutzt ist, haben wir es mißbraucht und dazu verwendet zwischen 40 und 80 Spur Minilaufwerken zu unterscheiden.

Byte 2 gibt die Gaplänge für den UPD 765 an, hierbei kann man folgende Faustregel verwenden: 128 Byte/Sek. 07h, 256 Byte/Sek. 0Eh, 512 Byte/Sek. 1Bh und 1024 Byte/Sek. 35h.

Die unteren 3 Bits des dritten Bytes der DISK\$TYPE\$LIST dienen zum Einstellen der Writeprecompensation für das betreffende Format (siehe hierzu c't 8/84: PROF-80)

Die restlichen 4 Bytes zeigen auf die Adressen des zugehörigen Disk Parameter Blocks und der zu verwendenden Sektor-Skew-Tabelle.

Der Disk Parameter Block (DPB)

Während die DISK\$TYPE\$LIST hauptsächlich für die physikalischen Eigenschaften eines Diskettenformats zuständig ist und nur BIOS-intern Verwendung findet, dient der DPB dazu, dem BDOS mitzuteilen, wie es die Diskette logisch verwalten soll. Hierzu zählen unter anderem Anzahl der Directory-Einträge, Diskettenkapazität, Blockgröße oder Anzahl der Systemspuren. Wer ein eigenes Diskettenformat in den PROF-80 "einbauen" will, muß deshalb auch noch den zugehörigen DPB einfügen. Ein DPB besteht aus 17 Bytes (Bild 4). Hierbei bedeuten:

SPT Mit diesem 2 Byte Wert wird die Anzahl der logischen 128 Byte Sektoren pro Spur angegeben. Mit der Anzahl der Sektoren ist also nicht die physikalische Anzahl pro Spur gemeint, sondern die gesamte Zahl der Bytes pro Spur geteilt durch 128.

Beim PROF-80 dient SPT gleichzeitig dazu, anzugeben, ob die Diskette ein- oder zweiseitig beschrieben werden kann. Wird nämlich die doppelte Anzahl der Sektoren/Spur eingetragen, dann geht PROF-80 davon aus, daß es sich um eine zweiseitige Diskette handelt und erweitert jede Spur mit der Spur auf der Rückseite. Beispiel: Bei einer 8" Double Density Diskette mit 1024 Byte/Sektor und 8 Sektoren pro Spur und Seite muß SPT gleich 64 gesetzt werden, falls die Diskette einseitig benutzt werden soll; setzt man SPT dagegen auf 128, dann wird die Diskette zweiseitig beschrieben.

BSH, Die Werte von BSH und BLM bestimmen zusammen die Blockgröße auf der Diskette. Die Blockgröße stellt ein vielfaches eines Sektors dar und ist die kleinste Einheit, die vom Directory zugewiesen werden kann. Das Directory kann jeder Datei immer nur eine ganze Anzahl von Blöcken zuweisen, d.h. im Mittel bleibt pro Datei etwa eine halbe Blockgröße ungenutzt. Je kleiner also die Blockgröße desto höher ist die Ausnutzung der Diskette, desto größer ist aber auch der Platzbedarf des Directory's; die Zugriffszeit auf die einzelnen Dateien

erhöht sich ebenfalls. Die nachfolgende Tabelle gibt einen Überblick über die Beziehung zwischen Blockgröße und BSH und BLM:

Blockgröße	BSH	BLM
1024	3	7
2048	4	15
4096	5	31
8192	6	63
16384	7	127

EXM Der Wert von EXM hängt von der Blockgröße und von DSM ab. Es gelten für EXM die folgenden Werte:

Blockgröße	DSM < 256	DSM > 255
1024	0	nicht möglich
2048	1	0
4096	3	1
8192	7	3
16384	15	7

DSM DSM bestimmt die maximale Speicherkapazität einer Diskette. Das Produkt  $(DSM+1)$  mal Blockgröße gibt die gesamte Kapazität der Diskette (ohne Systemspuren) in Bytes an.

DRM gibt die Anzahl der möglichen Directory Einträge weniger Eins an. Beispiel: Sollen 128 Einträge vorgesehen werden, dann gilt  $DRM=127$ .

CKS gibt die Größe des "Directory Check Vektors" an. Es gilt:  $CKS=(DRM+1)/4$ .

ALO, Auch das Directory selbst belegt Blöcke auf der Diskette AL1 und zwar von Block 0 an aufwärts, je nach Anzahl der Directory Einträge. Die 16 Bits von ALO,AL1 geben die Blöcke an, die vom Directory belegt werden. Für jeden

belegten Block wird eine '1' in AL0-AL1 geschrieben und zwar von links nach rechts auffüllend (Bild 5). Jeder Directory Eintrag belegt 32 Byte auf der Diskette, hat man z.B. 128 Directory Einträge und eine Blockgröße von 2048 Byte, dann werden 2 Blöcke für das Directory benötigt; AL0 hat dann den Wert C0h und AL1 hat den Wert 00h.

OFF Diese Feld gibt die Anzahl der Systemspuren an.

PSH, Diese beiden Felder geben zusammen die physikalische PHM Sektorgröße an:

Sektorgröße	PSH	PHM
128	0	0
256	1	1
512	2	3
1024	3	7
2048	4	15
4096	5	31

A n h a n g

---

logischer Kanal	kein Jumper	J4/1-3	J4/2-4	J4/3-5	J4/4-6
CONIN	GRIP	DUPLEX	USER1	USER2	GRIP
CONOUT	GRIP	DUPLEX	USER1	USER2	GRIP
AUXIN	DUPLEX	DUPLEX	DUPLEX	DUPLEX	DUPLEX
AUXOUT	DUPLEX	DUPLEX	DUPLEX	DUPLEX	DUPLEX
LST	SPOLER	SIMPLX	SIMPLX	SIMPLX	SIMPLX

Tabelle 1. Zuordnung der Ein-/Ausgabe-Kanäle nach dem Booten in Abhängigkeit von J4

---

---

Die DISK\$TYPE\$LIST dient dazu, ein Diskettenformat zu erkennen und Adressen des DPB und der Sektor-Skew-Tabelle sowie Parameter für die Floppy-Bedienung bereit zustellen. Ein Eintrag besteht aus insgesamt 8 Byte und ist folgendermaßen aufgebaut:

Byte 0: Disk Typ; die einzelnen Bits haben folgende Bedeutung:

Bit 0-1: Diese 2 Bits geben die Sektorgröße an.  
0=128 Byte, 1=256 Byte  
2=512 Byte, 3=1024 Byte

Bit 2-4: immer 0

Bit 5: 0=Maxi-Format, 1=Mini-Format

Bit 6: 0=Single Density, 1=Double Density

Bit 7: Diese Bit muß mit dem Pegel des Two-Sided Sided-Signals des UPD 765 übereinstimmen.

Byte 1: Anzahl der physikalischen Sektoren/Spur

Byte 2: Gap Länge für den UPD 765

Byte 3: Die unteren 3 Bits dieses Bytes werden vor jedem Diskzugriff auf die Eingänge P0-P2 des 9229 gegeben

Byte 4-5: Adresse des DPB für dieses Format

Byte 6-7: Adresse der Sektor-Skew-Tabelle für dieses Format

Tabelle 3: Aufbau eines Eintrags in der DISK\$TYPE-LIST

---

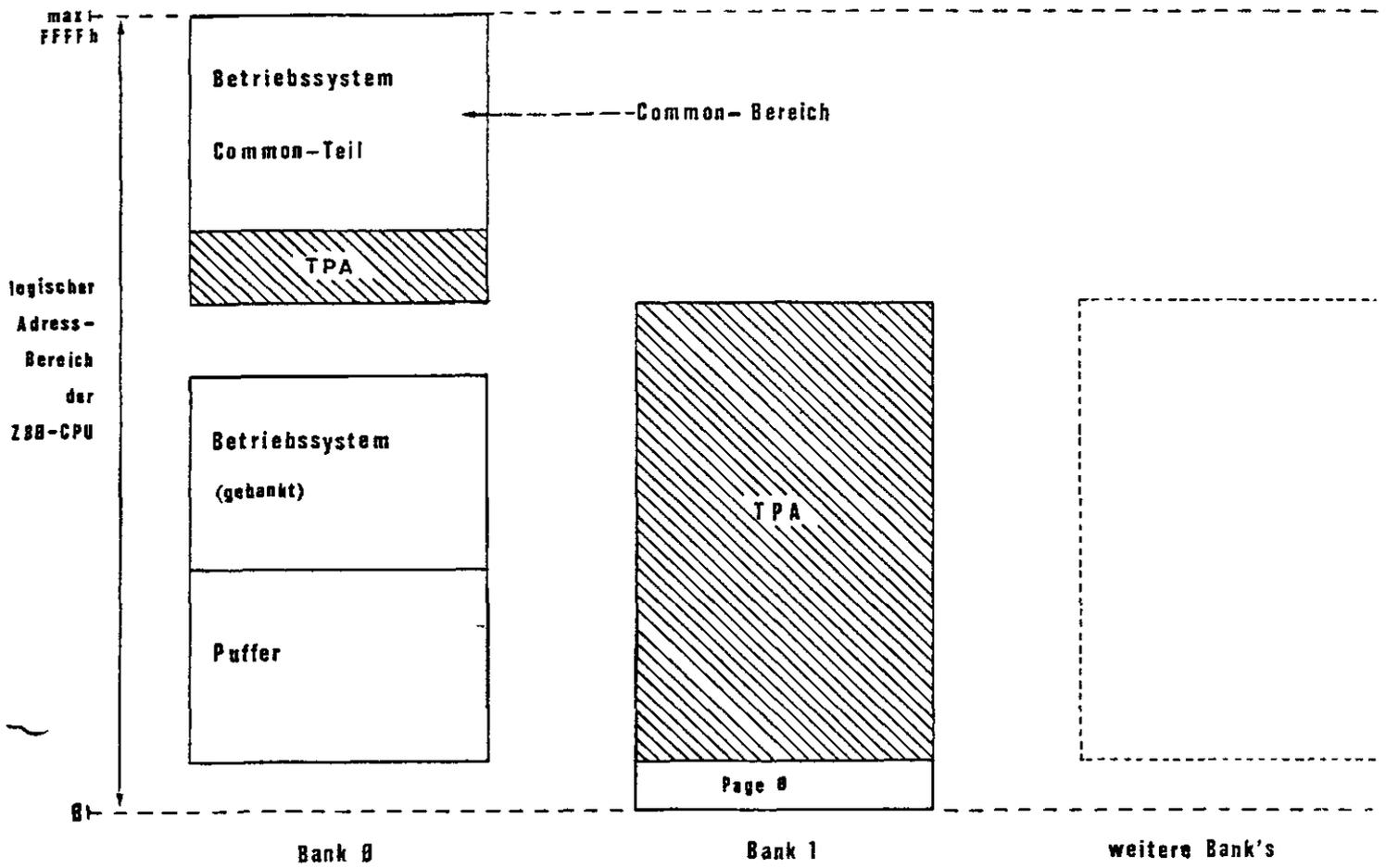


Bild 1. Allgemeine Speicherstruktur eines CP/M plus Systems

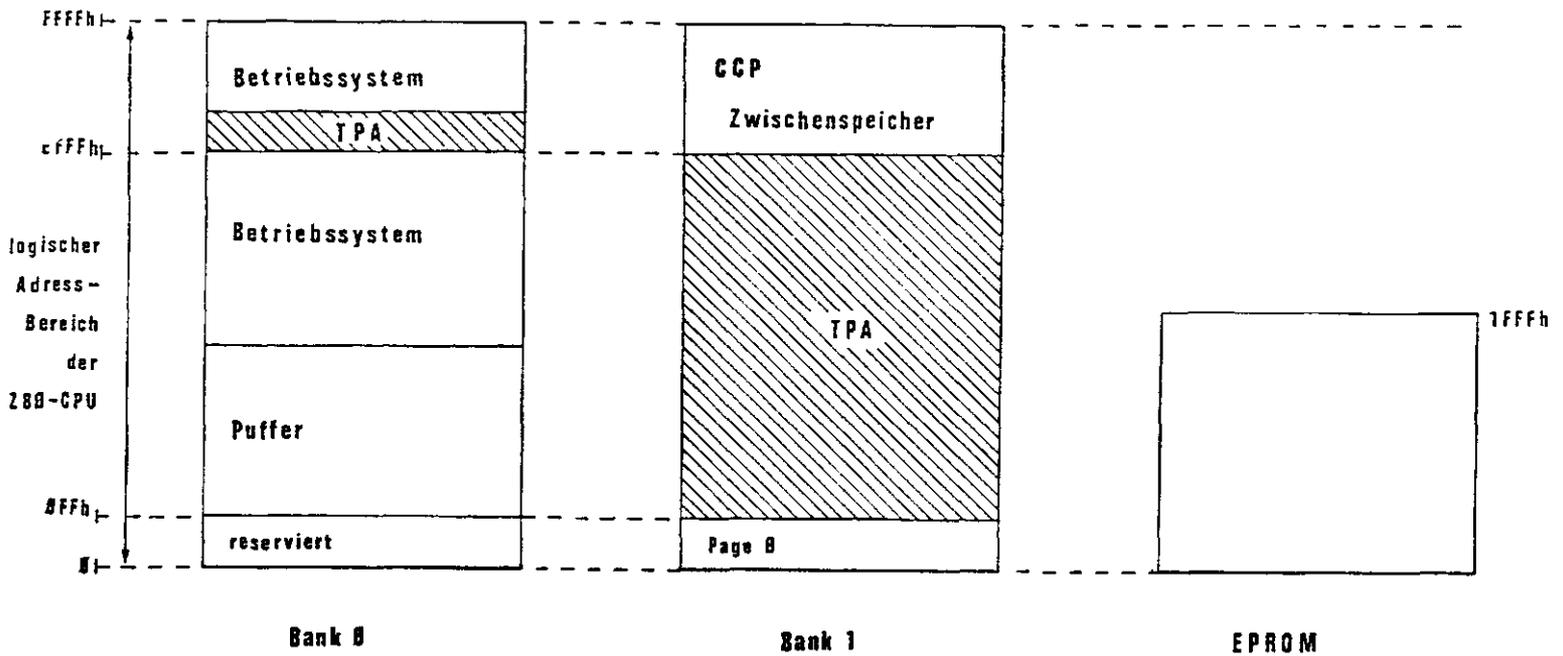


Bild 2. CP/M plus Speicheraufteilung im PRDF-85

---

Nr	Einsprung	Beschreibung
0	JMP BOOT	; Kaltstart Initialisierung, Laden des CCP usw.
1	JMP WBOOT	; Warmstart Initialisierung, CCP neu laden
2	JMP CONST	; prüfe ob Zeichen am Konsolenkanal anliegt
3	JMP CONIN	; lese ein Zeichen vom Konsolenkanal
4	JMP CONOUT	; gebe ein Zeichen an den Konsolenkanal aus
5	JMP LIST	; gebe ein Zeichen an den Druckerkanal aus
6	JMP AUXOUT	; gebe ein Zeichen an den Zusatzkanal aus
7	JMP AUXIN	; lese ein Zeichen vom Zusatzkanal
8	JMP HOME	; positioniere auf Spur 0 des ausgewählten Laufwerks
9	JMP SELDSK	; wähle ein Laufwerk aus
10	JMP SETTRK	; setze Spur beim ausgewählten Laufwerk
11	JMP SETSEC	; setze Sektor beim ausgewählten Laufwerk
12	JMP SETDMA	; setze Speicheradresse für nachfolgenden Diskzugriff
13	JMP READ	; lese den angewählten physikalischen Sektor
14	JMP WRITE	; beschreibe den angewählten physikalischen Sektor
15	JMP LISTST	; prüfe, ob Druckerkanal bereit ist ein Zeichen auszugeben
16	JMP SECTRN	; Sektor Umwandlungsroutine
17	JMP CONOST	; prüfe, ob Konsolenkanal bereit ist ein Zeichen auszugeben
18	JMP AUXIST	; prüfe, ob ein Zeichen am Zusatzkanal anliegt
19	JMP AUXOST	; prüfe, ob der Zusatzkanal bereit ist ein Zeichen auszugeben
20	JMP DEVTBL	; lade Adresse der Zeichen E/A-Tabelle
21	JMP DEVINI	; initialisiere Zeichen E/A-Treiber
22	JMP DRVTBL	; lade Adresse der Laufwerkstabelle
23	JMP MULTIO	; setze Anzahl der Sektoren
24	JMP FLUSH	; leere Deblocking Bu-fer (nur bei eigenem Deblocking notwendig)
25	JMP MOVE	; kopiere Speicherbereiche
26	JMP TIME	; lese/setze Echtzeituhr
27	JMP SELMEM	; schalte Speicherbank um
28	JMP SETBNK	; wähle Speicherbank für nachfolgenden Disk-Read/Write-Befehl aus
29	JMP XMOVE	; setze Speicherbanke für Interbank-MOVE
30	JMP USERF	; reserviert für Systemspezifische Funktionen
31	JMP RESERV1	; reserviert für zukünftige Erweiterungen
32	JMP RESERV2	; reserviert für zukünftige Erweiterungen

Bild 3. BIOS-Einsprung-Vektor

---

---

```

-----
/ SPT /BSH/BLM/EXM/ DSM / DRM /ALO/AL1/ CKS / OFF /PSH/PHM/
-----
16b  8b  8b  8b   16b   16b  8b  8b   16b   16b   8b  8b

```

Bild 4: Aufbau des Disk Parameter Blocks (DPB). 16b bedeutet 16 Bit Wort, 8b bedeutet Byte.

---



---

```

-----
I      ALO                I      AL1                I
-----
I I I I I I I I I I I I I I I I I
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15

```

Bild 5: Aufbau des AL-Vektors. Für jeden vom Directory belegten Block muß eine '1' von links beginnend eingetragen werden.

---

### **Anderung des Skewfaktors bei den Formaten 2 und 4**

Ab der BIOS-Version 23.10.1984 wurde der Skewfaktor bei zweiseitigen Minidisketten von 1 auf 2 geändert. Durch diese Änderung beschleunigt sich der Diskettenzugriff etwa um den Faktor 3.

Achtung der Datenaustausch mit älteren BIOS-Versionen kann bei Minidisketten nur noch über einseitig formatierte Disketten erfolgen.

### **Anschluß von 80-Spur Minilaufwerken**

Damit 80-Spur Minilaufwerke richtig verwaltet werden können, muß das Two-Sided-Signal am Laufwerksstecker auf Low liegen. Dies läßt sich am einfachsten durch einen Jumper zwischen Pin 10 und Pin 9 des Maxifloppy-Steckers (N4) erreichen oder mit dem 'U'-Befehl.